What is claimed is:

1. Apparatus for workload balancing in an asynchronous messaging system comprising:

means for obtaining the average depth of a queue of messages; and

means for controlling the number of server instances retrieving messages from the queue based on the average queue depth and one or more predetermined thresholds.

2. The apparatus of claim 1 comprising:

means, responsive to determining that the average queue depth exceeds a first predetermined threshold, for starting another server instance for retrieving messages from the queue.

3. The apparatus of claim 1 comprising:

means, responsive to determining that the average queue depth exceeds a first predetermined threshold, for requesting that another entity start a server instance for retrieving messages from the queue.

4. The apparatus of claim 3, wherein the controlling means comprises:

means for placing a trigger message on an initialisation queue, the trigger message being destined for the other entity, the trigger message indicating to the other entity that a server instance is to be started.

5. The apparatus of claim 4, wherein the apparatus is for use in a messaging system having a plurality of queues and the trigger message includes information regarding (i) which server to instantiate; and (ii) which queue the newly instantiated server instance should retrieve messages from.

6. The apparatus of claim 2 comprising:

means, responsive to determining that the first threshold has been exceeded, for resetting the average queue depth to less than the first threshold.

7. The apparatus of claim 1, wherein the means for controlling the number of server instances comprises:

means for terminating a server instance when the average queue depth falls below a second predetermined threshold.

8. The apparatus of claim 7 comprising:

means, responsive to determining that the average queue depth is below the second threshold, for resetting the average queue depth to be greater than the second threshold.

9. The apparatus of claim 7, wherein the means for terminating a server instance comprises at least one of:

(i) means for spoofing the server instance into believing that there are no more messages on the queue for it to process;

(ii) means for spoofing the server instance into believing that a queue manager, controlling the queue, is shutting down;
(iii)means for spoofing the server instance into believing that operator intervention has requested that the server instance shuts down; and
(iv) means for requesting that the server instance shuts down.

10. The apparatus of claim 7, wherein the means for terminating a server instance comprises:
means for requesting that another entity terminate the server instance.

11. The apparatus of claim 1 comprising:
means for setting a maximum number of server instances that can be active at any one time.

12. The apparatus of claim 1, comprising:
means for setting a minimum number of server instances that should be active at any one time.

13. The apparatus of claim 1 wherein the means for obtaining the average queue depth comprises:
means for calculating the queue's average depth.

14. The apparatus of claim 13 wherein the means for calculating comprises:
means for calculating a time weighted mean average queue depth.

15. The apparatus of claim 13, wherein the means for calculating comprises:

means for calculating an exponentially smoothed average queue depth.

16. An asynchronous messaging system for workload balancing comprising:

a queue comprising messages for processing by at least one server instance;

means for obtaining the average depth of the queue of messages;

means for controlling the number of server instances for retrieving such messages from the queue based on the average depth of the queue and one or more predetermined thresholds.

17. A server instance for processing messages from a queue, the server instance comprising:

means for obtaining the average depth of the queue of messages;

means for controlling the number of additional server instances based on the average depth of the queue and one or more predetermined thresholds.

18. The server instance of claim 17, wherein the controlling means comprises:

means for spawning an additional server instance when the average queue depth exceeds a first predetermined threshold.

19. The server instance of claim 17, wherein the controlling means comprises:

means for terminating an additional server instance when the average depth of the queue falls below a second predetermined threshold.

20. A method for workload balancing in an asynchronous messaging system comprising:

obtaining the average depth of a queue of messages; and

controlling the number of server instances retrieving messages from the queue based on the average queue depth and one or more predetermined thresholds.

21. The method of claim 20 comprising:

responsive to determining that the average queue depth exceeds a first predetermined threshold, starting another server instance for retrieving messages from the queue.

22. The method of claim 20 comprising:

responsive to determining that the average queue depth exceeds a first predetermined threshold, for requesting that another entity start a server instance for retrieving messages from the queue.

23. The method of claim 22, wherein the controlling step comprises:

placing a trigger message on an initialisation queue, the trigger message being destined for the other

entity, the trigger message indicating to the other entity that a server instance is to be started.

24. The method of claim 23, wherein the method is for use in a messaging system having a plurality of queues and the trigger message includes information regarding (i) which server to instantiate; and (ii) which queue the newly instantiated server instance should retrieve messages from.

25. The method of claim 21 comprising:

responsive to determining that the first threshold has been exceeded, resetting the average queue depth to less than the first threshold.

26. The method of claim 20, wherein the controlling step:

terminating a server instance when the average queue depth falls below a second predetermined threshold.

27. The method of claim 26 comprising:

responsive to determining that the average queue depth is below the second threshold, resetting the average queue depth to be greater than the second threshold.

28. The method of claim 26, wherein the step of terminating a server instance comprises at least one of:

(i) spoofing the server instance into believing that there are no more messages on the queue for it to process;
(ii) spoofing the server instance into believing that a queue manager, controlling the queue, is shutting down;
(iii)spoofing the server instance into believing that operator intervention has requested that the server instance shuts down; and
(iv) requesting that the server instance shuts down.

29. The method of claim 26, wherein the step of terminating a server instance comprises:
requesting that another entity terminate the server instance.

30. The method of claim 20 comprising:
setting a maximum number of server instances that can be active at any one time.

31. The method of claim 20, comprising:
setting a minimum number of server instances that should be active at any one time.

32. The method of claim 20, wherein the step of obtaining the average queue depth comprises:
calculating the queue's average depth.

33. The method of claim 32 wherein the step of calculating comprises:

calculating a time weighted mean average queue depth.

34. The method of claim 32, wherein the step of calculating comprises:

calculating an exponentially smoothed average queue depth.

35. A computer program for workload balancing in an asynchronous messaging system, the computer program comprising program code means adapted to perform the steps of:

obtaining the average depth of a queue of messages; and

controlling the number of server instances retrieving messages from the queue based on the average queue depth and one or more predetermined thresholds.